UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/527,812	11/29/2005	Christophe Justin Evrard	550-619	4576

23117 7590 09/08/2010
NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203

EXAMINER	
VICARY, KEITH E	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
09/08/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte CHRISTOPHE JUSTIN EVRARD and JULIE-ANNE FRANCOISE MARIE PRUVOST

Appeal 2009-005011
Application 10/527,812
Technology Center 2100

Before JOSEPH L. DIXON, JEAN R. HOMERE, and JAMES R. HUGHES, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON REQUEST FOR REHEARING[1]

---

[1] The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the "MAIL DATE" (paper delivery mode) or the "NOTIFICATION DATE" (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

STATEMENT OF THE CASE

In a paper filed December 29, 2009, Appellants request a rehearing under 37 C.F.R. § 41.52 from the Opinion of the Board of Patent Appeals and Interferences (hereinafter Board), dated November 12, 2009. In the Opinion, we affirmed the Examiner's rejection of claims 1 through 10. Appellants allege that the Board erred by misapprehending or overlooking numerous arguments presented in the Brief: (i) by misconstruing the claimed invention (Req. Reh'g. 2-4), (ii) by misconstruing the Examiner's findings (*id.* at 4-5), (iii) by misconstruing the issue on appeal (*id.* at 5-6), (iv) by misconstruing Qiu's teachings (*id.* at 6-8), and (v) by using the misconstrued issue as a guide in the analysis section to apply Qiu's misconstrued teachings to the misconstrued claims. (*Id.* at 8-13.) In particular, Appellants allege the following errors:

i. In describing Appellants' invention, the Board states that "[u]pon determining that the instruction result, when written, will not cause a change, the processor core processes the conditional instruction and subsequently stores the results thereof in a trash register." (*Id.* at 2.) This statement evidences the Board's misunderstanding of the claimed invention since it does not correspond to any language in the claim nor is it supported by the portions of Specification cited by the Board. (*Id.*) According to Appellants, the Board's description of the claimed invention fails to recite "at least one data processing instruction" always executed by a "processor core" wherein the data processing instruction is a "conditional-write data processing instruction encoding condition codes." (*Id.* at 3.)

ii. The Board misconstrues the Examiner's Findings by asserting that the Examiner finds Qiu teaches a "conditional-write data processing instruction encoding conditions codes specifying conditions." (*Id.* at 4-5.)

iii. The Board misconstrues the issue on appeal by omitting therefrom a critical element of the claim, whereby "at least one of the instructions executed by the processor core 'is a conditional-write data processing instruction encoding conditions codes.'" (*Id.* at 5-6.)

iv. The Board misconstrues Qiu's teachings in finding (FF. 3) that the reference discloses a value bit that determines whether or not a multiplication is likely to occur. (*Id.* at 6-7.) Further, Appellants allege that the Board's findings of fact fails to provide any disclosure in Qiu that teaches (1) a "conditional write data processing instruction encoding condition codes" and (2) "condition codes specifying conditions under which said conditional-write data processing instruction will or will not be permitted to write data to effect a change in state of said processor core." (*Id.*)

v. The Board applied Qiu's misconstrued teachings to the misconstrued claim in the analysis section based on a misconstrued issue, and ignored the central feature in Appellants' claim whereby the conditional-write instruction encodes condition codes. (*Id.* at 8-9.)

We have carefully reviewed the Opinion in light of Appellants' allegations of errors. We will address those remarks in the order in which they are presented in the Request, and as outlined above.

i. We find without merit Appellants' first allegation of error that we misconstrued the claimed invention. In particular, claim 1 recites in relevant parts:

> (a) at least one data processing instruction executed by said processor core is a conditional-write data processing instruction encoding condition codes specifying conditions under which said conditional-write *data processing instruction will or will not be permitted to write data to effect a change in state of said processor core*;
>
> (b) a trash register to which a result data value will be written instead of a data processing register upon execution of said conditional-write data processing instruction *when said condition codes within said conditional-write data processing instructions that do not permit a write to effect a change in state of said processor core*. (Emphasis Added)

We find that the plain meaning of the cited portions of the disputed claim is as follows:

(a) The processor core executes a conditional-write data processing *instruction having codes that specify the conditions* under which the *instruction will or will not be permitted to write data that causes a change in the state of the computer core*.

(b) Upon executing the instruction, writing result data value in a trash register instead of the data processing register when *the codes within the executed instruction do not permit a write to cause a change in the state of the computer core*.

Clearly from the above claim construction, Appellants' claimed invention does require that, upon determining that the instruction will not permit a write to cause a change in the processor core, writing a result data value in a trash register instead of a data processing register, as set forth in the original Opinion. (Op. 2.) Further, as set forth in the analysis section in the Opinion, the conditional-write instruction implies an instruction that embeds therein certain conditions under which writing to the processor will be permitted. (*Id.* at 7-9.) We reiterate our initial position that an ordinarily skilled artisan would readily understand that the cited language requires certain conditions to be coded within the instruction to thereby specify when to or not to allow a write to cause a change in the processor core. Therefore, we find that the claim recitation of "an instruction encoding condition codes" was sufficiently captured in our summary of the claimed invention. Accordingly, we find no error in our summary of Appellants' claimed invention.

ii. We find unavailing Appellants' second allegation of error that we misconstrued the Examiner's findings. In the last paragraph at page 12 bridging page 13 of the Examiner's Answer, the Examiner specifically discusses that Qiu's disclosure teaches conditional write-instructions, wherein the instructions have certain conditions associated therewith to determine whether to allow a write to cause a change to the state of the processor. Consequently, we find no error in our statement of the Examiner's findings as set forth in the original Opinion. (Op. 4-5.)

iii. We find unpersuasive Appellants' third allegation of error that we omit an essential claim limitation from the issue on appeal. The issue recites in relevant part "upon executing *a conditional-write data processing instruction when condition codes associated therewith* do not permit a write to effect a change in the processor core." (Op. 5.) We find this language to be substantially identical to the disputed claim language. While we chose to couch the issue in terms of the condition codes being "associated with" the instruction instead of the codes being "within" or "encoded" in the instruction, we find that Appellants' attempt to belabor this point is a red-herring. As discussed above, we find our use of the terms conditional-write instruction in the issue statement suffices to indicate that the instruction embeds codes that specify certain conditions. Furthermore, we find that the codes encoded within the instructions are also associated therewith. Consequently, we find no error in our statement of the issue as set forth in the original Opinion.

iv. We find Appellants' fourth allegation of error regarding Fact Finding 3 to be misplaced and irrelevant since we did not rely upon it in our analysis. Rather, we relied upon Fact Finding 4, which Appellants admit to be accurate. (Req. Reh'g at 7.) We will further discuss in the following paragraph the merits of Fact Finding 4 as it pertains to Appellants' claim.

v. We find unpersuasive Appellants' fifth allegation of error that our analysis ignored the claim limitation of a conditional instruction encoding condition codes. In fact, in the original Opinion, we expressly devoted two full pages of the analysis section to claim construction to fully ascertain the

meaning and scope of "conditional write data processing instruction" and "condition codes." (Op. 7-9.) In particular, we construed a "*conditional-write data processing instruction*" as a "command … to write to memory upon the occurrence of a predetermined condition." (Op. 9.) Appellants' Request for Rehearing conveniently fails to address this extensive claim construction since it clearly refutes their arguments. We find that by belaboring the difference between an instruction having condition codes associated therewith (i.e. an instruction that was coded to specify certain conditions) and an instruction encoding condition codes (i.e. an instruction having encoded therein codes specifying certain conditions), Appellants are merely splitting hairs. The ordinary skill artisan would readily appreciate that in order for any instruction or command to have any functionality, it must have been programmed accordingly. (i.e. encoding certain codes.) Thus, in this case, since the instruction is a conditional write instruction, it must have been programmed or encoded to specify certain conditions to permit writing to the memory of the data processor core. Hence, we reiterate our initial position that we are satisfied that Qiu's disclosure of storing an emulated multiplication result in the dummy register is conditioned upon the processor receiving a command to perform an unnecessary operation which will not change the state of the processor (FF. 4) teaches the disputed limitations. (Op. 9-10.) Consequently, we maintain our initial position that Qiu anticipates independent claim 1.

CONCLUSION

In view of the foregoing discussion, we deny Appellants' Request for Rehearing.

REQUEST FOR REHEARING-DENIED

peb

NIXON & VANDERHYE, PC
901 NORTH GLEBE ROAD, 11TH FLOOR
ARLINGTON, VA 22203